

LAPORAN PRAKTIKUM

ALGORITMA

PEMROGRAMAN

“PERCABANGAN”

DISUSUN OLEH:

ARYAHYAHUL FIKRA

2511532026

DOSEN PENGAMPU:

Dr. WAHYUDI, S.T, M.T

ASISTEN PRAKTIKUM:

RAHMAD DWIRIZKI



DEPARTEMEN INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

PROGRAM STUDI INFORMATIKA

UNIVERSITAS ANDALAS

2025

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa karena atas rahmat dan karunia-Nya, Laporan praktikum ini dapat diselesaikan dengan baik. Laporan praktikum ini disusun sebagai salah satu tugas dalam rangka memenuhi tugas yang diberikan oleh dosen dan juga asisten praktikum.

Padang, 2 Oktober 2025

Aryahiyahul fikra

DAFTAR ISI

DEPARTEMEN INFORMATIKA	1
KATA PENGANTAR	1
DAFTAR ISI.....	2
BAB I.....	1
PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Tujuan	2
1.3 Manfaat Pratikum	2
BAB II PEMBAHASAN.....	3
2.1 Uraian Program yang Dipraktikkan.....	3
2.1.1 Program NamaBulan.java	3
2.1.2 Program Nilai.java.....	3
2.1.3Program ifelse1.java	4
2.1.4 Program latif1.java	4
2.1.5 Program multiIf.java.....	5
2.2 Penjelasan Langkah Kerja.....	6
2.3 Analisis Hasil dan Teori	6
BAB III PENUTUP	8
3.1 Kesimpulan	8
3.2 Saran	8
Daftar Pustaka	9

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi yang pesat dalam beberapa dekade terakhir telah memberikan dampak besar terhadap berbagai aspek kehidupan manusia, termasuk di bidang pendidikan, industri, dan komunikasi. Salah satu pilar utama dalam perkembangan teknologi tersebut adalah **pemrograman komputer**. Di antara banyak bahasa pemrograman yang ada, **Java** merupakan salah satu bahasa yang sangat populer dan banyak digunakan di berbagai belahan dunia.

Java adalah bahasa pemrograman tingkat tinggi yang bersifat **object-oriented**, dikembangkan oleh Sun Microsystems pada tahun 1995, dan kini dikelola oleh Oracle. Keunggulan utama Java terletak pada kemampuannya untuk berjalan di berbagai platform (platform-independent) berkat konsep **Write Once, Run Anywhere (WORA)** yang dimilikinya. Hal ini memungkinkan program yang dibuat dalam Java untuk dijalankan di berbagai sistem operasi tanpa perlu diubah ulang.

Selain itu, Java memiliki sintaks yang relatif mudah dipahami bagi pemula, dokumentasi yang lengkap, dan komunitas pengguna yang besar, sehingga sangat mendukung proses belajar dan pengembangan perangkat lunak. Java juga menjadi bahasa utama dalam pengembangan berbagai aplikasi, seperti aplikasi desktop, mobile (Android), web, hingga sistem enterprise berskala besar.

Dengan melihat pentingnya Java dalam dunia pemrograman dan aplikasinya yang luas, sangatlah relevan untuk mempelajari dasar-dasar bahasa pemrograman Java, baik dari segi teori maupun praktik. Oleh karena itu, laporan ini disusun untuk memberikan pemahaman yang lebih mendalam tentang bahasa Java, struktur dasarnya, serta penerapan konsep pemrograman berorientasi objek dalam bahasa tersebut.

1.2 Tujuan

1. Membantu mahasiswa memahami konsep dasar pemrograman Java melalui penerapan langsung.
2. Melatih kemampuan menulis, mengompilasi, dan mengeksekusi program dengan mengikuti aturan sintaksis Java.
3. Meningkatkan keterampilan dalam memecahkan masalah (problem solving) dengan pendekatan algoritmik
4. Membiasakan mahasiswa bekerja sistematis dalam menyusun laporan yang memuat analisis hasil praktikum
5. Menanamkan sikap teliti, disiplin, serta tanggung jawab dalam melaksanakan kegiatan laboratorium.

1.3 Manfaat Pratikum

1. Mahasiswa dapat memahami konsep dasar pemrograman secara praktis, bukan hanya teori.
2. Melatih keterampilan logika dan analisis dalam menyusun serta mengeksekusi kode.
3. Memberikan pengalaman langsung dalam menggunakan bahasa Java sebagai salah satu bahasa pemrograman populer.
4. Menjadi bekal awal untuk memahami materi pemrograman yang lebih kompleks pada praktikum berikutnya.
5. Membiasakan mahasiswa dengan proses debugging dan pemecahan masalah dalam pemrograman.

BAB II

PEMBAHASAN

2.1 Uraian Program yang Dipraktikkan

Pada praktikum ini, mahasiswa mempraktikkan beberapa program Java yang menerapkan struktur percabangan (if-else dan switch-case). Percabangan digunakan untuk membuat keputusan berdasarkan kondisi tertentu sehingga program dapat menampilkan output yang sesuai dengan input pengguna. Berikut uraian masing-masing program:

2.1.1 Program NamaBulan.java

Program ini meminta input angka (1–12) lalu menampilkan nama bulan menggunakan percabangan switch-case. Jika input tidak sesuai (selain 1–12), maka ditampilkan pesan "Angka tidak valid".

```
1 package Pekan4;
2
3 import java.util.Scanner;
4
5 public class namabulan {
6
7     public static void main(String[] args) {
8         Scanner scanner=new Scanner(System.in);
9         System.out.println("Masukkan Nama Bulan (1 - 12) =");
10        int bulan = scanner.nextInt();
11        switch (bulan) {
12            case 1:
13                System.out.println("Januari");
14                break;
15            case 2:
16                System.out.println("Februari");
17                break;
18            case 3:
19                System.out.println("Maret");
20                break;
21            case 4:
22                System.out.println("April");
23                break;
24            case 5:
25                System.out.println("Mei");
26                break;
27            case 6:
28                System.out.println("Juni");
29                break;
30            case 7:
31                System.out.println("Juli");
32                break;
33            case 8:
34                System.out.println("Agustus");
35                break;
36            case 9:
37                System.out.println("September");
38                break;
39            case 10:
40                System.out.println("Oktober");
41                break;
42            case 11:
```

Output

```
Masukkan Nama Bulan (1 - 12) =
8
Agustus
```

2.1.2 Program Nilai.java

Program ini menggunakan if-else-if untuk menentukan nilai huruf berdasarkan input angka (0–100)

Nilai $\geq 81 \rightarrow A$

Nilai $> 70 \rightarrow B$

Nilai $\geq 60 \rightarrow C$

Nilai $\geq 50 \rightarrow D$

Selain itu $\rightarrow E$

```
1 package Pekan4;
2
3 import java.util.Scanner;
4
5 public class nilai {
6
7     public static void main(String[] args) {
8         int nilai;
9         Scanner input=new Scanner(System.in);
10        System.out.print("Input nilai angka = ");
11        nilai=input.nextInt();
12        input.close();
13
14        if(nilai >=81) {
15            System.out.println("A");
16        } else if(nilai >=70) {
17            System.out.println("B");
18        } else if(nilai >=60) {
19            System.out.println("C");
20        } else if(nilai >=50) {
21            System.out.println("D");
22        } else {
23            System.out.println("E");
24        }
25
26
27
28    }
29
30 }
31
32
```

2.1.3 Program ifelse1.java

Program ini menggunakan if-else sederhana untuk menentukan kelulusan berdasarkan IPK.

Jika IPK > 2.75 , mahasiswa dinyatakan lulus sangat memuaskan. Jika tidak, maka tampil pesan "Anda Tidak Lulus".

```
1 package Pekan4;
2
3 import java.util.Scanner;
4
5 Input IPK anda = 2.65
6 Anda tidak lulus
7
8
9
10
11
12        System.out.println("Anda lulus sangat memuaskan dengan IPK "+IPK);
13    } else {
14        System.out.println("Anda tidak lulus");
15    }
16
17
18
19
20
21 }
22
```

Output else

2.1.4 Program latif1.java

Program ini hampir sama dengan ifelse1.java, tetapi hanya menggunakan if tunggal. Jika

IPK > 2.75 , maka ditampilkan "Anda Lulus Sangat Memuaskan", sedangkan jika tidak memenuhi kondisi maka tidak ada output tambahan.

```

1 package Pekan4;
2
3 import java.util.Scanner;
4
5 public class latif {
6
7     public static void main(String[] args) {
8         double IPK;
9         Scanner input=new Scanner(System.in);
10        System.out.print("Input IPK anda = ");
11        IPK=input.nextDouble();
12        input.close();
13        if (IPK>2.75) {
14            System.out.println("Anda lulus sangat memuaskan dengan IPK "+IPK);
15        }
16    }
17 }
18 }
19 }
20

```

Output jika bernilai sesuai yang diminta

```

Input IPK anda = 2.90
Anda lulus sangat memuaskan dengan IPK 2.9

```

Output jika tidak sesuai dengan program yang diminta

```

Input IPK anda = 2.65

```

2.1.5 Program multiIf.java

Program ini lebih kompleks karena menggunakan beberapa kondisi if independen. Input berupa umur dan status kepemilikan SIM C (y/n). Output program:

Umur ≥ 17 dan punya SIM \rightarrow "Anda Sudah dewasa dan boleh bawa motor".

Umur ≥ 17 dan tidak punya SIM \rightarrow "Anda Sudah dewasa tetapi tidak boleh bawa motor".

Umur < 17 dan tidak punya SIM \rightarrow "Anda Belum cukup umur bawa motor".

Umur < 17 dan punya SIM → "Anda Belum cukup umur punya SIM".

```
1 package Pekan4;
2
3 import java.util.Scanner;
4
5 public class multiif {
6
7     public static void main(String[] args) {
8         int umur;
9         char sim;
10        Scanner a=new Scanner(System.in);
11        System.out.print("Input umur anda = ");
12        umur= a.nextInt();
13        System.out.print("Apakah Anda Sudah Punya Sim C:");
14        sim=a.next().charAt(0);
15        a.close();
16        if((umur >= 17)&&(sim=='y')) {
17            System.out.println("Anda sudah dewasa dan boleh bawa motor");
18        }
19        if((umur >= 17)&&(sim!='y')) {
20            System.out.println("Anda sudah dewasa tetapi tidak boleh bawa motor");
21        }
22        if((umur < 17)&&(sim=='y')) {
23            System.out.println("Anda belum cukup umur bawa motor");
24        }
25        if((umur < 17)&&(sim=='y')) {
26            System.out.println("Anda belum cukup umur buat sim");
27        }
28    }
29
30 }
31
32 }
33
```

2.2 Penjelasan Langkah Kerja

Langkah kerja umum dari semua program adalah sebagai berikut:

1. Program diawali dengan deklarasi package dan import java.util.Scanner untuk membaca input dari pengguna.
2. Scanner digunakan untuk menangkap input angka maupun huruf.
3. Program memproses input menggunakan percabangan (if-else atau switch-case) sesuai logika yang ditentukan.
4. Hasil dari kondisi ditampilkan menggunakan System.out.println().
5. Scanner ditutup dengan close() untuk menghindari kebocoran resource.

2.3 Analisis Hasil dan Teori

Berdasarkan percobaan yang dilakukan, dapat dianalisis bahwa:

Struktur switch-case pada NamaBulan.java lebih efisien digunakan ketika terdapat banyak pilihan yang saling eksklusif, misalnya pemilihan bulan berdasarkan angka. Switch-case membuat kode lebih rapi dibanding if-else bertingkat.

Struktur if-else-if pada Nilai.java efektif digunakan untuk pengambilan keputusan dengan beberapa rentang nilai. Hal ini sesuai teori bahwa if-else cocok untuk kondisi yang saling berurutan (range).

If-else sederhana (ifelse1.java dan latif1.java) menunjukkan perbedaan implementasi:

ifelse1.java memberikan dua kemungkinan output (Lulus / Tidak Lulus).

latif1.java hanya menampilkan output jika kondisi benar, sedangkan jika salah tidak menampilkan apapun.

Multi-if (multiIf.java) digunakan untuk mengevaluasi lebih dari satu kondisi secara independen. Hal ini sesuai teori bahwa multi-if berbeda dengan if-else-if, karena semua kondisi akan diperiksa satu per satu, bukan hanya salah satunya.

Dari hasil percobaan, seluruh program berjalan sesuai dengan logika yang diharapkan. Kesimpulannya, percabangan dalam Java memberikan fleksibilitas dalam pengambilan keputusan, baik untuk kondisi sederhana maupun kondisi yang lebih kompleks.

BAB III

PENUTUP

3.1 Kesimpulan

Berdasarkan praktikum percabangan menggunakan bahasa pemrograman Java, dapat disimpulkan bahwa:

1. Struktur percabangan (if, if-else, if-else-if, switch-case) sangat penting dalam pemrograman untuk menentukan alur program berdasarkan kondisi tertentu.
2. Switch-case lebih cocok digunakan untuk pilihan tetap dan terbatas, seperti penentuan nama bulan dari angka.
3. If-else-if efektif digunakan pada kondisi yang berbentuk rentang nilai, misalnya penentuan nilai huruf berdasarkan skor angka.
4. If tunggal dan if-else sederhana dapat digunakan untuk keputusan cepat yang hanya membutuhkan satu atau dua kondisi.
5. Multi-if memberikan keleluasaan untuk mengecek beberapa kondisi secara independen, meskipun kode menjadi lebih panjang dibandingkan if-else-if.
1. Secara umum, hasil praktikum menunjukkan bahwa seluruh program dapat berjalan sesuai dengan logika yang diterapkan, dan mahasiswa dapat memahami perbedaan penerapan berbagai jenis percabangan.

3.2 Saran

Untuk pengembangan lebih lanjut, saran yang dapat diberikan adalah:

1. Program dapat ditambahkan validasi input agar pengguna tidak memasukkan data yang salah, misalnya huruf pada input angka.
2. Penggunaan nested if (percabangan bersarang) dapat dipraktikkan untuk mempelajari kasus yang lebih kompleks.
3. Program dapat dikembangkan dengan antarmuka grafis (GUI) agar lebih interaktif dan menarik.
2. 4. Perlu latihan lebih lanjut pada kasus nyata, misalnya sistem login, perhitungan biaya, atau penentuan status kelulusan agar pemahaman konsep percabangan lebih aplikatif.

Daftar Pustaka

Deitel, H. M. & Deitel, P. J. (2017). Java: How to Program. Pearson.

Schildt, H. (2018). Java: The Complete Reference. McGraw-Hill Education.

Oracle. (2023). Java Documentation. Diakses dari: <https://docs.oracle.com/javase/>

Wahana Komputer. (2019). Belajar Java untuk Pemula. Yogyakarta: Andi.

Sutarman. (2012). Pengantar Teknologi Informasi. Jakarta: Bumi Aksara.